**实验三 软件过程模型****、Scrum开发方法、可行性分析报告**

**夏思晓 202100031020**

**一、实验目的：**

1. 研讨传统软件开发过程模型与敏捷开发

2. 了解Scrum开发方法

3. 完成可行性分析报告

**二、实验内容：**

1. 小组分工讨论传统软件开发过程模型与敏捷开发（中几种主要方法）的比较，分析各自的优缺点，以及如何应用于自己的项目中？并且分析自己项目中可能存在的风险，细化风险管理（做出风险分级及应对预案）。

**（一）软件开发过程模型比较及应用**

表 1 软件开发过程模型比较

|  |  |  |
| --- | --- | --- |
| **软件开发过程模型** | **优点** | **缺点** |
| 传统模型 | 强调按照严格的阶段顺序进行开发，每个阶段都有明确的目标和输出。这有利于大型软件开发过程中人员的组织、管理，也有利于软件开发方法和工具的研究，从而提高大型软件项目开发的质量和效率。 | 1. 过分强调了分阶段实施，使得开发过程各个阶段之间存在严重的顺序性和依赖性。 2. 思维成果的可重用性很差。 3. 忽视了人在软件开发过程中的地位和作用。 4. 严格的流程和规范导致项目难以应对变化，尤其是在需求频繁变化的场景中。 5. 开发过程不可逆转，一旦需求发生变更，修改代价较大。客户往往不能给出全面需求，这会影响后面各阶段的工作。 |
| 瀑布模型 | 1. 为项目提供了按阶段划分的检查点。 2. 当前活动完成后，只需要去关注后续活动。 3. 它提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。 4. 阶段划分明确，文档化管理严格，适用于大型、长周期、稳定需求的项目。 | 1. 缺乏灵活性，不能反映实际的代码开发方式。 2. 无法及时验证某一软件开发活动是否正确。 |
| 原型模型 | 1. 从认知论的角度看，原型方法遵循了人们认识事物的规律，因而更容易为人们普遍接受。 2. 原型方法将模拟的手段引入分析的初期阶段，沟通了人们的思想，缩短了用户和开发人员之间的距离。 | 1. 对于一个大型系统，如果不经过系统分析得到系统的整体划分，而直接用原型来模拟是很困难的。 2. 对于原有应用的业务流程、信息流程混乱的情况，原型构造与使用有一定的困难。 3. 对于一个批处理系统，由于大部分活动是内部处理的，因此应用原型方法会有一定的困难。 4. 文档容易被忽略。 5. 项目难以规划和管理。 |
| 增量模型 | 1. 有利于增加客户对系统的信心。 2. 降低系统失败风险。 3. 提高系统可靠性，稳定性和可维护性。 | 1. 增量粒度难以选择。 2. 把每个新的增量构建集成到现有软件体系结构中时，必须不破坏原来已经开发出的产品。 3. 容易退化为边做边改模型，从而是软件过程的控制失去整体性。 |
| 螺旋模型 | 1. 支持用户需求的动态变化, 具有良好的可扩充性和可修改性。也支持软件系统的可维护性，每次维护过程只是沿螺旋模型继续多走一两个周期。 2. 原型易于用户和开发人员共同理解需求，还可作为继续开发的基础，并为用户参与所有关键决策提供了方便。 3. 螺旋模型为项目管理人员及时调整管理决策提供了方便，进而可降低开发风险。 | 1. 很难让用户确信这种演化方法的结果是可以控制的。 2. 建设周期长，而软件技术发展比较快，所以经常出现软件开发完毕后，和当前的技术水平有了较大的差距，无法满足当前用户需求。 |
| 敏捷开发模型 | 1. 采用简单计划策略，不需要长期计划和复杂模型，开发周期短。 2. 在全过程采用迭代增量开发、反馈修正和反复测试的方法，能够适应用户经常变化的需求。 3. 强调客户参与和反馈，能够更好地满足客户需求。 4. 敏捷开发方法最适合有较高不确定性的环境，能够快速响应需求变化。 | 1. 敏捷开发方法需要大量的培训和技能才能成功实施。 2. 需要一定程度的组织转型，包括业务用户与开发团队之间的信任和伙伴关系。 3. 注重人员的沟通，忽略文档的重要性，若项目人员流动大太，给维护带来不少难度。 4. 对编码人员的经验要求高，若项目存在新手比较多时，老员工比较累。 |

在“职等你来”人才招聘系统的开发过程中，我们将多种软件开发过程模型的思想相结合，并主要应用敏捷开发模型的思想，确保项目高效、灵活地响应快速变化的需求。具体应用包括：

**（1）用户故事（User Stories）**

在敏捷开发中，用户故事是捕获用户需求的有效方式。在“职等你来”项目中，可以定义一系列用户故事，如“作为求职者，我希望能够创建和保存简历”，“作为招聘者，我希望能够搜索和筛选合适的候选人”等。用户故事应该简短、明确，并且可以通过验收测试来验证是否完成。

**（2）迭代开发（Iterative Development）**

将整个项目分解为多个迭代（Sprint），每个迭代周期通常为2周至1个月。在每个迭代中，选择并开发一部分高优先级的用户故事。每个迭代结束时，都应该有一个可展示、可测试的产品增量。这样，即使在项目早期，用户也能看到系统的部分功能并提供反馈。

**（3）持续集成（Continuous Integration）**

使用持续集成工具（如Jenkins、Travis CI等）来自动化构建、测试和部署代码。每当有代码提交时，都会触发自动化流程，确保代码的质量。这有助于及早发现代码中的错误，并确保团队成员之间的代码能够顺利集成。

**（4）原型开发（Prototyping）**

在项目早期，可以制作低保真或高保真的原型来模拟系统的界面和功能。通过原型，用户可以直观地了解系统的使用方式，并提供反馈。原型开发可以与迭代开发相结合，在每个迭代中根据用户反馈进行改进和调整。

**（5）定期站会（Regular Stand-up）**

每天定时举行简短的站会，让每个团队成员都有机会分享他们的进展、遇到的问题以及接下来的计划。站会应该保持简短，通常不超过15分钟。通过站会，团队成员可以保持同步，并及时解决问题。

**（6）优先级排序（Prioritization）**

使用产品待办事项列表（Product Backlog）来管理所有的用户故事和任务。根据用户需求和价值，对任务进行优先级排序。在每个迭代开始时，从产品待办事项列表中选择最高优先级的用户故事进行开发。随着项目的进行，可以根据用户反馈和市场变化调整优先级。

**（7）结合其他模型思想**

在某些阶段，可以借鉴瀑布模型的思想，进行详细的系统设计和规划。这有助于确保系统的整体架构和稳定性。

增量模型的思想也可以用于项目的某些部分，如先实现系统的核心功能，然后逐步增加新的功能和模块。

螺旋模型的风险评估和管理方法可以在项目全过程中应用，以确保项目能够顺利应对各种风险和挑战。

**（二）风险管理**

**1、风险分析**

1. 技术风险

① 软件开发延期：招聘软件涉及的功能模块较多，如职位发布、简历筛选、在线沟通等，每个模块的开发都可能遇到技术难题，导致开发进度延误。

② 技术兼容性问题：招聘软件需要支持多种设备和浏览器，不同平台之间的兼容性测试可能耗费大量时间和资源。

③ 数据安全问题：招聘软件涉及用户隐私信息，如简历、联系方式等，如果安全措施不到位，可能导致数据泄露或被非法访问。

（2）市场风险

① 市场需求变化：随着行业发展，招聘市场可能对软件功能有新的需求，如视频面试、AI简历筛选等，若项目未能及时跟进，可能导致市场竞争力下降。

② 竞争对手：市场上已有众多招聘软件，竞争对手可能通过价格战、功能创新等方式抢占市场份额。

（3）团队风险

① 技能不足：团队成员可能缺乏某些关键技术或行业经验，影响软件的开发质量和效率。

（4）资源风险

① 资金短缺：项目初期可能面临资金紧张的问题，导致无法购买必要的硬件设备或软件工具。

② 硬件资源不足：随着项目的推进，可能需要更多的服务器和存储设备来支持软件运行和数据存储。

**2、风险分级**

根据项目的实际情况，可以按发生概率的高低，事件影响的大小将上述风险进行如下分级：

（1）高风险：软件开发延期、数据安全问题、市场需求变化。

（2）中风险：技术兼容性问题、竞争对手、技能不足。

（3）低风险：资金短缺、硬件资源不足。

**3、应对预案**

（1）技术风险应对

① 制定详细的项目计划：明确每个功能模块的开发进度和责任人，确保按时交付。

② 加强兼容性测试：针对不同设备和浏览器进行充分的测试，确保软件的稳定运行。

③ 加强数据安全管理：采用加密技术、访问控制等手段保护用户数据，定期进行安全检查和漏洞修复。

（2）市场风险应对

① 持续关注市场动态：通过市场调研、用户反馈等方式了解市场需求变化，及时调整产品功能和定位。

② 差异化竞争策略：在功能、用户体验等方面与竞争对手形成差异化，提升市场竞争力。

（3）团队风险应对

① 定期培训：组织技术培训和经验分享活动，提升团队成员的技能水平。

（4）资源风险应对

① 合理预算：制定详细的资金预算和使用计划，确保资金的合理使用。

② 提前规划硬件资源：根据项目需求提前采购和部署必要的服务器和存储设备。

1. 阅读Scrum开发方法文档，理解Scrum过程工作模型。

Scrum是一个用于产品开发或项目管理的轻量级框架，强调跨职能团队的协作、自组织、快速反馈和持续改进。Scrum将产品开发过程划分为一系列被称为Sprint的短周期迭代，每个Sprint通常不超过4周，最常见的是两周。在每个Sprint中，团队从产品待办事项列表中选择高优先级的事项进行开发，并通过每日Scrum会议、Sprint评审会议和Sprint回顾会议来监控进度、调整方向并持续改进。

**（1）Scrum中的角色**

Scrum中有三个核心角色：产品负责人（Product Owner）、开发团队（Development Team）和ScrumMaster。**产品负责人**负责确定产品特性和优先级，为团队提供清晰的目标和方向。产品负责人需要与开发团队紧密合作，确保产品待办事项列表的有效管理。**开发团队**由跨职能的团队成员组成，负责在每个Sprint中按照优先级顺序完成产品待办事项列表中的事项。开发团队是自组织的，并对Sprint的交付结果负责。**ScrumMaster**负责保护Scrum过程和团队免受外部干扰，确保Scrum的顺利实施。ScrumMaster的角色是服务型领导，帮助团队解决冲突、提高协作效率。

**（2）Scrum的工件**

Scrum中有三个主要工件：产品待办事项列表（Product Backlog）、Sprint待办事项列表（Sprint Backlog）和增量产品（Increment）。**产品待办事项列表**包含了产品所需的所有特性和功能，按照优先级进行排序。产品负责人负责维护和管理这个列表。**Sprint待办事项列表**包含了团队在当前Sprint中计划完成的事项。在Sprint计划会议中，开发团队从产品待办事项列表中选择事项，并预估完成这些事项所需的工作量。**增量产品**是每个Sprint结束时团队交付的可工作产品。增量产品必须是可用的，并且满足Sprint目标的要求。在Sprint评审会议中，团队向利益关系人展示增量产品，并获取反馈。

**（3）Scrum的事件**

Scrum中有四个主要事件：Sprint计划会议（Sprint Planning）、每日Scrum会议（Daily Scrum）、Sprint评审会议（Sprint Review）和Sprint回顾会议（Sprint Retrospective）。**Sprint计划会议**在Sprint开始前举行，用于确定Sprint的目标和计划。**每日Scrum会议**每天举行的简短会议，用于同步团队工作进度、识别障碍并调整计划。**Sprint评审会议**在Sprint结束时举行，用于展示增量产品、收集反馈并讨论下一步计划。**Sprint回顾会议**在Sprint评审会议之后举行，用于总结Sprint中的经验教训、识别改进点并制定改进措施。

3. 完成可行性分析报告。

**下周五（含）前将可行性分析报告提交给相应的助教**

将小组工作文档、进度等文档保存到每个小组选定的协作开发平台上